Describing “shape” (feature extraction)

INF 3300/4300 Lecture 6

Asbjørn Berge

Lecture outline

- What is shape?
- Input to shape descriptors
- Description from contours
- Description from regions
- Summary and short Matlab demo

Abstracting objects

- We have available techniques for low-level image description
  - Segmentation algorithms: contigous regions of an image with similar properties
    - Encoded as "labeled" image, runlenght code etc
  - Contour descriptors: tracing of the border between regions with differing properties
    - Usually encoded as a 1D-representation, signature, chaincode etc
- The challenge is to describe the **shape** of these regions or contours in a machine readable format
  - Numeric features - recognize by statistical models
  - Syntactical description - recognize by (fuzzy) rules

Applications of ”shape” descriptors

- Several application areas in the field of image analysis is dependent on shape description
  - Object recognition
  - Image retrieval
  - Computer vision
  - Video compression (MPEG-7 and MPEG-4)

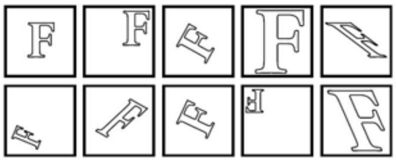
What is ”shape”?

- No general accepted methodology of shape description
- Humans use adjectives (”rounded, spiky, elongated”) or templates (”egg shaped”)
- What happens if we rotate or scale the object? (Your idea of ”pear shaped” is still the same)
- Depending on application, the descriptions can be simplified with heuristics
- As an example, the most salient information about an object is usually in areas with high curvature (where the boundary ”turns sharply”)

Considerations

- Input representation form, boundaries or whole regions?
- Object reconstruction ability?
- Incomplete shape recognition ability?
- Local/global description?
- Mathematical or heuristic techniques?
- Statistical or syntactic object description?
- Robustness of description to translation, rotation, and scale transformations?
- Shape description properties in different resolutions?
  - description changes discontinously.
- Robustness against
  - noise
  - geometric sampling (discretization)
  - quantization

What kind of robustness is needed?



- •Translation invariance
- •Scale invariance
- •Rotation invariance, but what about 6 and 9?
- •Reflection invariance
- •Warp invariance

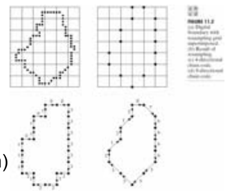
Region identification

- Goal of segmentation was to achieve complete segmentation, now, the regions must be labeled.
- Search image pixel for pixel, and sequentially number each foreground pixel you find according to the labeling of its neighbors.
- Label collision is a very common occurrence - examples of image shapes experiencing this are U-shaped objects, mirrored E objects, etc.
- The algorithm is basically the same in 4-connectivity and 8-connectivity, the only difference being in the neighborhood mask shape.
- Result of region identification is a matrix the same size as the image with integers representing each pixels region label.
- This description of regions will be the input to our shape descriptors
- Matlab function implementing this is *bwlabel*

Figure 6.3 *Masks for region identification: (a) In 4-connectivity, (b) in 8-connectivity, (c) label collision.*

Contour representation

- Goal of contour methods is to describe contours in images
- Hopefully, our contour detection method delivers a sequence of pixel coordinates in the image!
- The pixel coordinates is not necessarily rectangular (cartesian) coordinates
  - Polar coordinates from a reference point (usually image origin)
  - Chain code and a starting point



Descriptors from the contour

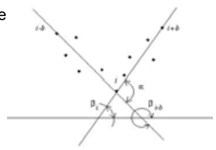
- Boundary length
- Area
- Curvature
- Bending energy
- Signature
- Basis expansion (Fourier)

Descriptors from the contour

- Boundary length
  - Simple to derive from chain code
  - Count 1 for each horz-vert move, and √2 for each diagonal move
  - Distance measure differs when using 8- or 4-neighborhood
- Area
  - Can be calculated from the boundary by Greens theorem
  - Surface integral equals boundary integral
    $A = \int\int_S dxdy = \int_C xdy$
  - Simple to implement, follow the contour, x and dy follow from pixels in the sequence

Descriptors from the contour

- Curvature
  - In the continous case, curvature is the rate of change of slope.
    $|\kappa(s)|^2 = \left[\frac{d^2x}{ds^2}\right]^2 + \left[\frac{d^2y}{ds^2}\right]^2$
  - In the discrete case, difficult because boundary is locally ragged.
  - Use difference between slopes of adjacent boundary segments to describe curvature at point of segment intersection.
  - Curvature can be calculated from chain code.



Descriptors from the contour

- Bending energy
  - From physics: if the curve is a rod, how much energy is needed to bend it into its shape?
  - Sum of squared curvature over a length L, $\text{BE} = \frac{1}{L}\sum_{s=1}^{L}\kappa^2(s)$
  - Simple to calculate from curvature estimate from chain code

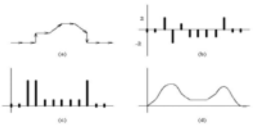
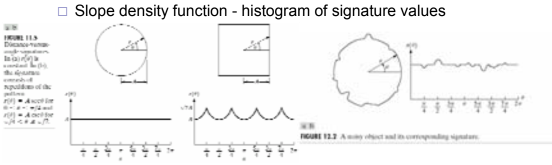


Figure 6.8 Bending energy: (a) Chain code 0, 0, 2, 0, 1, 0, 7, 6, 0, 0, (b) curvature 0, 2, -2, 1, -1, -1, -1, 2, 0, (c) sum of squares gives the bending energy, (d) smoothed version.

Descriptors from the contour

- Signatures
  - 1D function representation of the boundary
  - Distance from the boundary as a function of angle
  - Alternative: angle between contour tangent and reference line as function of position at contour
  - Rotation and scale variant
  - Slope density function - histogram of signature values



Fourier descriptors

- Idea - boundary can be viewed as 1D periodic signal
- Perform a forward Fourier transform of the signal

$$F(u) = \frac{1}{M}\sum_{k=0}^{M-1} f(x)\exp\left(\frac{-2\pi iuk}{M}\right)$$

for $u \in [0, M-1]$

- F(0) now contains the center of mass of the object, and the coefficients F(1), F(2), F(3), ..., F(M − 1) will describe the object in increasing detail.
- These features depend on rotation, scaling and starting point on the contour.
- We do not want to use all coefficients as features, but terminate at F(N), N < M. This corresponds to setting F(k) = 0, k > N − 1

Fourier descriptors

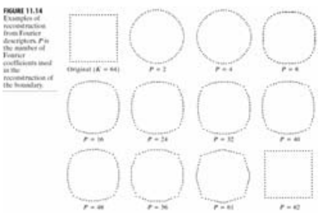
- When transforming back, we get an approximation to the original contour

$$\hat{f}(k) = \sum_{u=0}^{N-1} F(u)\exp\left(\frac{2\pi iuk}{M}\right)$$

- defined for $k \in [0, M-1]$
- We have only used N features to reconstruct each component of $f(k)$, but k still runs from 0 to $M-1$.
- The number of points in the approximation is the same (M), but the number of coefficients (features) used to reconstruct each point is smaller ($N < M$).
- The first 10 - 15 descriptors are found to be sufficient for character description.
- The Fourier descriptors can be invariant to translation and rotation if the boundary description is appropriately chosen

Fourier descriptors demo

- Fourier descriptor applet



Region descriptors on contours

- A contour can be viewed as a hull of a region
- This means that many descriptors meant for regions can also be used on contours
- Especially statistical moments are simple to modify for such uses
- We have already seen statistical moments, in the estimate of area covered

Region descriptors

- Much of the region description methodology focus on *moments*
  - Borrowed ideas from physics and statistics
- We will define the grayscale moment and derive heuristic region descriptors from this
- Moments can be robustified and made invariant
- Some of the heuristics are not directly related to the moments
  - Perimeter - the length of the contour of the region
  - Compactness - ratio between the squared perimeter and area

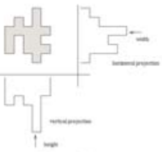


Figure 6.22 Projections.

Moments

- For a given intensity distribution *g(x, y)* we define moments
- m_{pq} by

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q g(x, y) dx dy$$

- For sampled (and bounded) intensity distributions f (x, y)

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$$

- A moment m_{pq} is of *order* p + q.
- For binary images, where
- f (x, y) = 1 → object pixel
- f (x, y) = 0 → background pixel

Moments

- Area

$$m_{00} = \sum_x \sum_y f(x, y)$$

- Center of mass

$$m_{10} = \sum_x \sum_y x f(x, y) = \bar{x} m_{00} \quad \Rightarrow \quad \bar{x} = \frac{m_{10}}{m_{00}}$$

$$m_{01} = \sum_x \sum_y y f(x, y) = \bar{y} m_{00} \quad \Rightarrow \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

Grayscale moments

- In gray scale images, where *f(x,y)* ∈ *[0,...,G-1]* we may regard *f(x,y)* as a discrete 2-D probability distribution over (x,y)
- We should then have

$$m_{00} = \sum_x \sum_y f(x, y) = 1$$

- And if this is not the case we can normalize by

$$F(x, y) = f(x, y)/m_{00}$$

Grayscale moments

- The total intensity is now

$$m_{00} = \sum_x \sum_y f(x, y)$$

- Center of mass has coordinates $(\bar{x}, \bar{y})$

$$m_{10} = \sum_x \sum_y x f(x, y) = \bar{x} m_{00} \quad \Rightarrow \quad \bar{x} = \frac{m_{10}}{m_{00}}$$

$$m_{01} = \sum_x \sum_y y f(x, y) = \bar{y} m_{00} \quad \Rightarrow \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

- Or, if we use the normalized image, F(x, y)

$$m_{10} = \sum_x \sum_y x F(x, y) = \bar{x}$$

$$m_{01} = \sum_x \sum_y y F(x, y) = \bar{y}$$

Central moments

- These are position invariant moments

$$\mu_{p,q} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

- where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \qquad \bar{y} = \frac{m_{01}}{m_{00}}$$

- The total intensity and the center of mass are given by

$$\mu_{00} = \sum_x \sum_y f(x, y), \qquad \mu_{10} = \mu_{01} = 0$$

- This corresponds to computing ordinary moments after having translated the object so that center of mass is in origo.
- Central moments are independent of position, but are not scaling or rotation invariant.

Variance

- The two second order central moments measure the spread of points around the centre of mass
  $\mu_{20} = \sum_x \sum_y (x-\bar{x})^2 f(x,y)$
  $\mu_{02} = \sum_x \sum_y (y-\bar{y})^2 f(x,y)$
- statisticans like to call these measurements variance, while physicists will use the term moments of inertia. However, the pointspread might not be perfectly aligned with the coordinate axes, and thus we get a cross moment of inertia
  $\mu_{11} = \sum_x \sum_y (x-\bar{x})(y-\bar{y}) f(x,y)$
- and this is what statisticians call covariance or correlation.
- Orientation of the object can be derived from these moments, which means that they are not invariant to rotation.

Skew and Kurtosis

- Can we measure if the points in the region are spread evenly around the mean?
- Yes, the third order central moments μ_{03}, μ_{30}, μ_{21}, μ_{12} measure that! This is commonly referred to as *skew* and is a measure of the symmetry of the pointspread
- Furthermore, the fourth order central moments μ_{04}, μ_{40}, μ_{31}, μ_{13}, μ_{22}, μ_{22} are referred to as *kurtosis*. While this is actually a measure of "fatness of the tails" if you ask a statistician, you can think of it as an overall measure of even distribution of points.
- Note that when we increase the order of the moments, more moments are needed to describe the region. Why is that?

More uses for moments

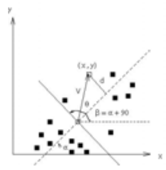
- If the region R is an ellipse with center in the origin, R = {(x, y)|$dx^2 + 2exy + fy^2 \le 1$} where d, e, f determine the lengths of the major and minor axes and orientation.
- There is a relationship between this ellipse and the second order moments μ such that

$$\begin{bmatrix} d & e \\ e & f \end{bmatrix} = \frac{1}{4(\mu_{20}\mu_{02} - \mu_{11}^2)} \begin{bmatrix} \mu_{20} & -\mu_{11} \\ -\mu_{11} & \mu_{02} \end{bmatrix}$$

- This relationship enables us to measure the orientation of the region as well as its eccentricity.

More uses for moments

- The orientation of an object is commonly defined as the angle relative to the first coordinate axis for which a line through the centroid has the least moment of inertia
- A statistician will call this the principal component, and not surprisingly this also correspond to the principal axis of the ellipse
- This direction can be found by minimizing the moment of inertia around a rotated axis



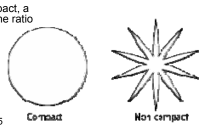
$$(\mu|\alpha) = \sum_{\{x,y\} \in R} d^2 = ((x-\overline{x})*cos\beta + (y-\overline{y})*sin\beta)^2$$

Orientation, eccentricity and compactness

- By deriving this equation and setting to zero, orientation can be shown to be
  $\alpha = \frac{1}{2}\tan^{-1}\left[\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}\right]$

- In fact there will be two extrema for , a minimum and a maximum, exactly 90◦ apart
- So we can find the major and the minor axes of the best fitting ellipse to the region
- The **eccentricity** is defined as the ratio of the ellipse axes
  $\varepsilon = \sqrt{1 - \frac{b^2}{a^2}}$

- Eccentricity is not accurate if the object is circular and compact, a moment based measure of **compactness** of the region is the ratio of the area and the variance in the axes,
  $c = \frac{\mu_{00}}{\mu_{20}+\mu_{02}}$



But what if the regions are rotated, scaled or otherwise mutilated?

- Scale: Since we know that the area of the region is μ_{00} we can just divide area out of the moment generating function
  $\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\gamma}}, \quad \gamma = \frac{p+q}{2} + 1, \quad p+q \ge 2.$
- Rotation: Rotate our coordinate system such that the correlation μ_{11}=0 - i.e. rotate it to correspond with the ellipse axes
  - No simple equation for normalization
  - Hu-moments implement this idea
- Rotation, scaling, shear and translation - aka affine transforms:
  - Flusser et al found four moments that are invariant under affine transforms
- Note that higher order moments are increasingly affected by noise (since the coordinate values are amplified by the exponent)

Moments from the contour

- Assume a closed boundary as an ordered sequence z(i) of Euclidean distance between the centroid and all N boundary pixels.
- Contour sequence moments can be estimated as

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)]^r$$

$$\mu_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)-m_1]^r$$

Moments from the contour

- Translation, rotation, and scale invariant one-dimensional normalized contour sequence moments:

$$\bar{m}_r = \frac{m_r}{(\mu_2)^{r/2}} = \frac{\frac{1}{N}\sum_{i=1}^{N}[z(i)]^r}{\left[\frac{1}{N}\sum_{i=1}^{N}[z(i)-m_1]^2\right]^{r/2}}$$

$$\bar{\mu}_r = \frac{\mu_r}{(\mu_2)^{r/2}} = \frac{\frac{1}{N}\sum_{i=1}^{N}[z(i)-m_1]^r}{\left[\frac{1}{N}\sum_{i=1}^{N}[z(i)-m_1]^2\right]^{r/2}}$$

- Since less samples are used for our estimates, the impact of variation in "individual" samples are higher (Thus these moment descriptors are somewhat noise sensitive)

What about using another basis?

- Just as the contour can be represented on a Fourier basis, regions can be mapped on a orthogonal set of complex (two dimensional) polynomials
- The Zernike basis (again(!) "stolen" from physics) has been very popular in OCR
- Zernike polynomials are orthogonal on a unit circle.

$$A_{nm} = \frac{n+1}{\pi}\sum_x\sum_y f(x,y)\,[V_{nm}(x,y)]^*,$$

where $x^2+y^2 \leq 1$

- The magnitudes $|A_{nm}|$ are rotation invariant.

Zernike moments

- The Zernike moments are projections of the input image onto a space spanned by the orthogonal V functions

$$V_{nm}(x,y) = R_{nm}e^{jm\tan^{-1}(y/x)}$$

where j = √−1, n ≥ 0, |m| ≤ n, n − |m| is even, and

$$R_{nm}(x,y) = \sum_{s=0}^{(n-|m|)/2}\frac{(-1)^s(x^2+y^2)^{(n/2)-s}(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!}$$

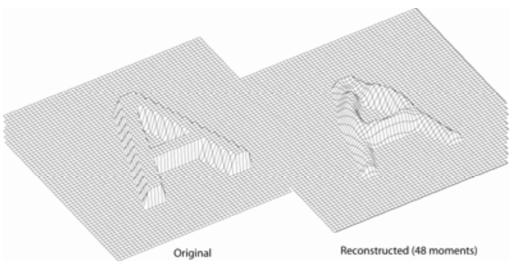
Zernike moments

- The image within the unit circle may be reconstructed to an arbitrary precision by

$$f(x,y) = \lim_{N\to\infty}\sum_{n=0}^{N}\sum_m A_{nm}V_{nm}(x,y)$$

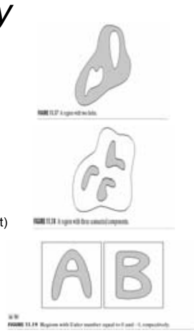
where the second sum is taken over all |m| ≤ n, such that n − |m| is even.

Zernike moments



A side note : *Topology*

- This is a group of warp invariant integer features.
- Some require thinning to obtain the object skeleton.
- Some are based on the convex hull and convex deficiency of a
- region
  - Number of holes in the object
  - Number of terminations (one line from a point)
  - Number of breakpoints or corners (two lines from a point)
  - Number of branching points (three lines from a point)
  - Number of crossings (more than three lines from a point)
  - Number of components (conn.)
  - Euler number, E = C − H (number of components - holes)



Matlab hints for lecture and exercise

- For the exercise you need to be familiar with
  - Thresholding images (already covered)
  - Matlab (IP toolbox) functions
    - *bwlabel*
    - regionprops
  - This lecture, of course! ☺

Matlab demo

Locate and measure ”roundness” and ”area” of the objects in this image